Law Offices of
SCHNECK & SCHNECK

Thomas Schneck
David M. Schneck
Gina McCarthy
Nissa Strottman

P.O. BOX 2-E
SAN JOSE, CALIFORNIA 95109-0005

80 S. Market Street
Third Floor
San Jose, California 95113-2303

Email: mail@patentvalley.com

Telephone: (408) 297-9733

Facsimile: (408) 297-9748

Patents and Trademarks

August 25, 2003

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Re: Certified Copy of Priority Document
U.S. Serial No.: 10/615,475
Filed: July 7, 2003
For: EFFICIENT MULTIPLICATION SEQUENCE FOR LARGE INTEGER OPERANDS WIDER THAN THE MULTIPLIER HARDWARE
Our ref: ATM-214 (V. Dupaquis et al.)

Dear Sir:

Transmitted herewith for the above-identified patent application is a certified copy of the priority document, French application no. 03/04299 filed April 7, 2003.

Respectfully submitted,

Thomas Schneck

Reg. No. 24,518

Schneck & Schneck
P.O. Box 2-E
San Jose, CA 95109-0005
(408) 297-9733

CERTIFICATE OF MAILING

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Signed: Merle P. Garcia
Typed Name: Merle P. Garcia

Date: August 25, 2003

Encl: Certified copy of priority document
cc: J. McGuire, Esq. w/encl.

R E P U B L I Q U E F R A N C A I S E

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 27 JUIN 2003

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

Martine PLANCHE

INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE
SIEGE
26 bis, rue de Saint Petersbourg
75800 PARIS cedex 08
Téléphone : 33 (0)1 53 04 53 04
Télécopie : 33 (0)1 53 04 45 23
www.inpi.fr

ETABLISSEMENT PUBLIC NATIONAL CREE PAR LA LOI N° 51-444 DU 19 AVRIL 1951

1er dépôt

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 33 (1) 53 04 53 04 Télécopie : 33 (1) 42 94 86 54

BREVET D'INVENTION
CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle - Livre VI

cerfa N° 11354*02

REQUÊTE EN DÉLIVRANCE
page 1/2

BR1

Cet imprimé est à remplir lisiblement à l'encre noire 08 540 @ W / 010801

Réservé à l'INPI

REMISE DES PIÈCES
DATE
LIEU 7 AVRIL 2003
75 INPI PARIS
N° D'ENREGISTREMENT NATIONAL ATTRIBUÉ PAR L'INPI 0304299
DATE DE DÉPÔT ATTRIBUÉE PAR L'INPI 0 7 AVR. 2003

1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE

BREESE-MAJEROWICZ
3 avenue de l'Opéra
75001 PARIS

Vos références pour ce dossier *(facultatif)* 33312/FR

Confirmation d'un dépôt par télécopie	☐ N° attribué par l'INPI à la télécopie
2 NATURE DE LA DEMANDE	**Cochez l'une des 4 cases suivantes**
Demande de brevet	☒
Demande de certificat d'utilité	☐
Demande divisionnaire	☐
Demande de brevet initiale	N° Date
ou demande de certificat d'utilité initiale	N° Date
Transformation d'une demande de brevet européen *Demande de brevet initiale*	☐ N° Date

3 TITRE DE L'INVENTION (200 caractères ou espaces maximum)

SÉQUENCE DE MULTIPLICATION EFFICACE POUR OPÉRANDES À GRANDS NOMBRES ENTIERS PLUS LARGES QUE LE MATÉRIEL MULTIPLICATEUR

4 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE FRANÇAISE	Pays ou organisation Date N° Pays ou organisation Date N° Pays ou organisation Date N° ☐ **S'il y a d'autres priorités, cochez la case et utilisez l'imprimé «Suite»**
5 DEMANDEUR (Cochez l'une des 2 cases)	☒ **Personne morale** ☐ **Personne physique**
Nom ou dénomination sociale	Atmel Corporation
Prénoms	
Forme juridique	constituée selon les lois de l'État du Delaware
N° SIREN	
Code APE-NAF	
Domicile ou siège — Rue	2325 Orchard Parkway
Domicile ou siège — Code postal et ville	SAN JOSE California 95131
Domicile ou siège — Pays	U.S.A.
Nationalité	U.S.A.
N° de téléphone *(facultatif)*	N° de télécopie *(facultatif)*
Adresse électronique *(facultatif)*	
	☐ **S'il y a plus d'un demandeur, cochez la case et utilisez l'imprimé «Suite»**

Remplir impérativement la 2ème page

1er dépôt

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

BREVET D'INVENTION
CERTIFICAT D'UTILITÉ

REQUÊTE EN DÉLIVRANCE
page 2/2

BR2

Réservé à l'INPI

REMISE DES PIÈCES
DATE 7 AVRIL 2003
LIEU 75 INPI PARIS
N° D'ENREGISTREMENT NATIONAL ATTRIBUÉ PAR L'INPI 0304299

DB 540 @ W / 010801

V s références pour ce dossier : *(facultatif)*		33312/FR
6 MANDATAIRE *(s'il y a lieu)*		
Nom		BREESE
Prénom		Pierre
Cabinet ou Société		BREESE-MAJEROWICZ
N °de pouvoir permanent et/ou de lien contractuel		
Adresse	Rue	3 avenue de l'Opéra
	Code postal et ville	7 5 0 0 1 Paris
	Pays	France
N° de téléphone *(facultatif)*		01 47 03 67 77
N° de télécopie *(facultatif)*		01 47 03 67 78
Adresse électronique *(facultatif)*		office@breese.fr
7 INVENTEUR (S)		**Les inventeurs sont nécessairement des personnes physiques**
Les demandeurs et les inventeurs sont les mêmes personnes		☐ Oui ☒ Non : **Dans ce cas remplir le formulaire de Désignation d'inventeur(s)**
8 RAPPORT DE RECHERCHE		**Uniquement pour une demande de brevet (y compris division et transformati n)**
Établissement immédiat ou établissement différé		☒ ☐
Paiement échelonné de la redevance *(en deux versements)*		**Uniquement pour les personnes physiques effectuant elles-mêmes leur propre d´pôt** ☐ Oui ☐ Non
9 RÉDUCTION DU TAUX DES REDEVANCES		**Uniquement pour les personnes physiques** ☐ Requise pour la première fois pour cette invention *(joindre un avis de non-imposition)* ☐ Obtenue antérieurement à ce dépôt pour cette invention *(joindre une copie de la décision d'admission à l'assistance gratuite ou indiquer sa référence)* : **AG**
Si vous avez utilisé l'imprimé «Suite», indiquez le nombre de pages jointes		

10 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE
(N m et qualit´ du signatair)
BREESE Pierre

921038

VISA DE LA PRÉFECTURE OU DE L'INPI

La loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés s'applique aux réponses faites à ce formulaire.
Elle garantit un droit d'accès et de rectification pour les données vous concernant auprès de l'INPI.

SÉQUENCE DE MULTIPLICATION EFFICACE POUR OPÉRANDES À GRANDS NOMBRES ENTIERS PLUS LARGES QUE LE MATÉRIEL MULTIPLICATEUR

La présente invention concerne des circuits de traitement arithmétique, spécifiquement un matériel multiplicateur, et les procédés d'utilisation de celui-ci pour exécuter des opérations de multiplication ou de multiplication-accumulation (et les opérations associées d'élévation au carré) impliquant au moins un opérande de plusieurs mots qui est plus large que le matériel multiplicateur.

L'invention concerne en particulier la commande matérielle de la séquence de multiplication pour exécuter de telles opérations à plusieurs mots d'une manière efficace, où le procédé est caractérisé par l'ordre particulier dans lequel sont traités les mots des opérandes.

Le matériel de multiplication a nécessairement une taille finie, typiquement définie comme ayant une paire d'entrée d'opérandes à mot unique et une sortie résultats en deux mots. Afin d'exécuter des opérations de multiplication-accumulation, la sortie du multiplicateur est normalement reliée à un circuit accumulateur, qui à au moins une largeur de deux mots plus un bit. (Le bit supplémentaire peut faire partie du résultat ou simplement être présent en tant qu'information de REPORT indiquant soit un dépassement par valeurs supérieures dans le cas d'une addition ou d'un dépassement par valeurs inférieures dans le cas de soustraction dans la partie accumulation de l'opération.) L'opération de base

est donc $R = Z \pm XY$. Pour une multiplication simple, $R = XY$, l'entrée accumulateur $Z = 0$. Pour les opérations d'élévation au carré, $X = Y$. L'opération de base est habituellement conçue pour exécuter une arithmétique d'entiers standard, mais le matériel de multiplication qui exécute l'arithmétique polynomiale existe également, notamment pour utiliser les applications cryptographiques.

En cryptographie et dans un certain nombre d'autres applications, on a besoin de multiplier de très gros nombres entiers comprenant un grand nombre de mots. Pour effectuer ces opérations à l'aide d'opérandes qui sont beaucoup plus larges que le matériel de multiplication, les opérandes doivent être découpés en une pluralité de segments à mot unique et introduits dans le matériel dans une certaine séquence spécifiée. Ces segments font l'objet d'opérations et les résultats intermédiaires sont accumulés de sorte que le produit final est calculé comme la somme de produits croisés de poids divers. Les segments d'opérande ayant la largeur d'un mot, ainsi que les résultats partiels, sont enregistrés dans une mémoire qui est adressée par le séquenceur d'opérations du matériel multiplicateur.

Une séquence type conserve un segment d'opérandes constant tandis que d'autres segments d'opérande sont analysés mot par mot pour être introduits dans le multiplicateur ; ensuite le premier opérande incrémente au segment large d'un mot suivant et l'analyse du deuxième opérande est répétée. Si $X = \sum_i x_i w^i$, $Y = \sum_j y_j w^j$, et $Z = \sum_k x_k w^k$, avec $w = 2^n$, alors

$$R = \Sigma_k r_k w^k = Z \pm XY = \Sigma_k z_k w^k \pm \Sigma_i \Sigma_j (x_i y_j) w^{i+j},$$

où i+j = k, et où n est la taille du mot en bits. Ainsi, dans une séquence d'opérations typique, les mots y_j sont passés sur tous les j pour un mot fixe x_i, puis i est incrémenté d'une unité et le cycle de mots y_i est répété pour le nouveau x_i.

Tandis que la nouvelle séquence décrite ci-dessus est directe, facile à programmer et donne le résultat correct, chaque étape ou cycle nécessite une moyenne de trois accès de la mémoire à accès direct. En particulier, chaque étape nécessite que y_j et z_k soient extraits de la mémoire, et qu'un résultat partiel r_k soit réécrit en mémoire.

Un objet de l'invention est de proposer une séquence de multiplication à plusieurs mots efficace pour les opérations à grands entiers, qui nécessite en moyenne un seul accès mémoire par multiplication.

L'objet est réalisé par un procédé qui traite la séquence de multiplication en groupes de deux poids-mot de résultat (appelés colonnes). Dans un groupe de paires de colonnes, la séquence procède par colonnes alternantes avec des poids de segment d'opérande augmentant constamment (ou diminuant constamment) (qualifiés de modèle en zigzag), de sorte que l'un des segments provenant d'un cycle de multiplication précédent est aussi utilisé dans le présent cycle de multiplication, et, sauf éventuellement pour le premier cycle de multiplication dans un groupe donné, seul un des segments d'opérande doit être extrait de la mémoire dans un cycle de multiplication quelconque pour ce groupe. Des additions de produits partiels de mêmes poids de résultats sont exécutées dans une opération accumulée qui

est transmise avec l'opération de multiplication. Des segments de deux mots d'un paramètre accumulé séparé peuvent aussi être ajoutés au début ou à la fin d'une séquence d'opérations accumulées du groupe correspondant.

Un module de multiplication accumulation (MAC) exécute les cycles de multiplication et d'accumulation comme ordonné par un séquenceur d'opérations préprogrammée dans le logiciel interne dans le matériel de multiplication, avec lectures des segments d'opérande depuis la mémoire à accès direct (RAM) transférées au module MAC par le biais de registres internes au matériel de multiplication. De même, les écritures des résultats intermédiaires et finaux sont ramenés dans le registre interne pour segments de paramètres accumulés et finalement retournés dans la RAM.

Une autre amélioration ajoute des registres de mémoire cache internes pour contenir les segments de paramètres fréquemment utilisés qui reviennent au début ou à la fin de chaque groupe de cycle de multiplication.

L'invention a un avantage sur les séquences de multiplication précédentes en ce qu'elle adapte largement les exigences des multiplicateurs pour les segments de paramètre à la largeur de bande d'accès mémoire d'une lecture et d'une écriture par cycle de multiplication, de sorte qu'une multiplication globale de grands entiers de plusieurs mots est exécutée efficacement en un minimum de cycles. L'opération de multiplication peut aussi être étendue sur plusieurs cycles (en utilisant le traitement "pipeline"), auquel cas on considère encore que, en moyenne, une multiplication est exécutée par cycle, puisqu'un résultat de multiplication est disponible par

cycle et puisqu'une nouvelle opération de multiplication peut être lancée par cycle.

Ainsi, l'invention concerne un procédé de fonctionnement d'un circuit de multiplication pour calculer le produit de deux opérandes (X et Y) dont au moins un d'eaux est sensiblement plus large que le circuit de multiplication, le circuit de multiplication ayant une paire d'entrées d'opérandes d'une largeur de deux mots et une sortie résultat d'une largeur de deux mots, où un mot est un nombre spécifié de bits, chacun des opérandes composé d'une pluralité de segments d'opérande contigus d'une largeur de mot ordonnée (x_k et y_m), caractérisés par des poids spécifiques (k et m), le circuit de multiplication ayant accès à une mémoire, le procédé comprenant les étapes consistant à :

- charger des segments d'opérandes d'une largeur d'un mot des deux opérandes dans un ordre spécifié à partir de la mémoire dans le circuit de multiplication, le circuit de multiplication comprenant au moins deux registres (RX et RY) ayant accès à ladite mémoire pour contenir temporairement les segments chargés ;
- multiplier les segments chargés pour obtenir des produits intermédiaires d'une largeur de deux mots, les produits intermédiaires ayant un poids égal à la somme des poids des segments chargés ;
- ajouter des produits intermédiaires du même poids dans un accumulateur, l'accumulateur ayant une taille de trois mots plus un nombre de bits de report, suffisante pour gérer une taille d'opérande maximale spécifiée, l'accumulateur connecté à un registre d'entrée de deux mots (RZ) pour contenir temporairement tout produit d'un poids spécifique ajouté précédemment et un registre de

sortie à deux mots (RR) pour contenir les résultats d'une étape d'addition, lesdits registres (RZ et RR) ayant accès à ladite mémoire ; et

- enregistrer les résultats accumulés depuis ledit registre de sortie (RR) de nouveau dans ladite mémoire au moins après accumulation de tous les produits intermédiaires du poids spécifié ;

dans lequel l'ordre spécifié pour les segments d'opérande de chargement dans lesdits registres est une séquence définie par les poids des produits intermédiaires résultants, dans lequel l'étape de multiplication est faite par groupes successifs de deux poids de produits adjacents, avec la séquence à l'intérieur d'un groupe sélectionnée de sorte que, à l'exception d'une première opération de multiplication dans un groupe donné, au plus un des segments d'opérande doive être lu de la mémoire et placé dans un registre (RX ou RY), l'autre segment d'opérande étant déjà enregistré dans l'autre registre (RY ou RX) à partir de l'opération de multiplication immédiatement précédente.

Avantageusement, le registre d'entrée de l'accumulateur (RZ) sera chargé avec un segment d'un mot accumulé (Z), où le segment est du même poids que les produits intermédiaires à former dans ladite étape de multiplication, moyennant quoi le procédé de fonctionnement exécute une opération de multiplication-accumulation sur une paire d'opérandes de multiplication et un opérande d'accumulation.

De préférence, la séquence se déroulera à l'intérieur d'un groupe dans un modèle en zigzag de poids de segments d'opérande augmentant constamment.

Avantageusement, la séquence se déroulera à l'intérieur d'un groupe dans un modèle en zigzag de poids de segments d'opérande diminuant constamment.

De préférence, le circuit de multiplication contiendra en outre un ensemble de registre de mémoire cache interne, lesdits registres de mémoire cache étant chargés depuis la mémoire avec des segments d'opérande qui sont fréquemment utilisés dans une opération de multiplication.

Avantageusement, la séquence se déroulera à l'intérieur d'un groupe commençant par une opération de multiplication avec au moins un segment d'opérande enregistré dans un registre de mémoire cache dans une première direction en zigzag de poids de segments d'opérande augmentant ou diminuant constamment, puis passe directement à une opération de multiplication du groupe pas encore exécutée qui a aussi au moins un segment d'opérande enregistré dans un registre de mémoire cache et continue dans une deuxième direction en zigzag de poids de segments d'opérande diminuant ou augmentant constamment jusqu'à ce que toutes les opérations de multiplication pour ce groupe soient terminées.

De préférence, l'ordre spécifié sera préprogrammé dans le logiciel interne dans un séquenceur d'opérations du circuit de multiplication, avec des longueurs de mot d'opérande incluses en tant que paramètre d'entrée pour une commande de multiplication.

L'invention se rapporte également à un circuit de multiplication, comprenant :

- un module de multiplication-accumulation (MAC) comprenant un ensemble multiplicateur qui introduit des segments d'opérandes récepteur à mot unique à multiplier

pour former un produit intermédiaire de deux mots, et comprenant aussi un circuit accumulateur avec une première entrée de deux mots pour recevoir des produits intermédiaires provenant du multiplicateur, une deuxième entrée de deux mots pour recevoir une valeur accumulée, et une sortie de trois mots plus un certain nombre de bits de report suffisant pour gérer une taille d'opérande maximum spécifiée pour proposer une somme de valeurs d'entrée provenant des entrées de deux mots, la sortie de l'accumulateur retournant aussi deux mots à la deuxième entrée de deux mots ;

- un ensemble de registres d'adresses internes pour adresser une mémoire à accès direct ;
- un ensemble de registres de données internes (RX, RY, RZ, RR) connectés pour recevoir et transmettre des segments desdites opérandes depuis et à destination de ladite mémoire et aussi connectés audit ensemble multiplicateur et audit accumulateur de manière à fournir des segments d'opérande aux entrées de ceux-ci et à en recevoir des résultats, chacun desdits segments d'opérande ayant un poids spécifié ; et
- un séquenceur d'opérations pour contrôler l'accès à la mémoire par lesdits adresse interne et registres de données et contrôler la séquence d'opérations de multiplication et d'accumulation par lesdits ensemble multiplicateur et accumulateur respectifs,

dans lequel ladite séquence est définie par les poids des produits résultants égaux à la somme des poids des segments d'opérande en cours de multiplication, dans lequel les opérations de multiplication sont faites dans des groupes successifs de deux poids de produits adjacents, avec la séquence dans un groupe sélectionné de

sorte que, excepté une première opération de multiplication dans un groupe donné, au maximum un des segments d'opérande doive être lu de la mémoire dans un registre (RX ou RY), l'autre segment d'opérande étant déjà enregistré dans l'autre registre (RY ou RX) depuis l'opération de multiplication immédiatement précédente.

Un mode d'exécution de l'invention sera décrit ci-après, à titre d'exemple non limitatif, avec référence aux figures annexées dans lesquelles :

La Figure 1 est une vue en projection horizontale schématique de l'architecture principale d'un système de traitement qui comprend un moteur de multiplication de la présente invention.

La Figure 2 est un diagramme d'interface représentant les registres et l'interface mémoire pour un moteur de multiplication typique selon la présente invention.

La figure 3 est une vue en projection horizontale schématique détaillée d'un module MAC dans le moteur de multiplication de la figure 1.

La figure 4 est un tableau de plusieurs multiplications et additions mot par mot établie par opérande et par poids de résultat servant à illustrer la séquence d'opération selon la présente invention.

Les figures 5a, 5b et 5c sont un tableau montrant une séquence d'opérations pour un mode de réalisation de multiplication de la présente invention, utilisant un ensemble additionnel de 5 registres de mémoire cache stockant des segments d'opérande fréquemment utilisés.

La figure 6 est un tableau de plusieurs multiplications et additions mot par mot disposées comme

sur la figure 4 par opérande et par poids de résultat pour une opération de multiplication-accumulation "rectangulaire" exemplaire avec des opérandes de différentes tailles.

Les figures 7a, 7b et 7c sont un tableau montrant un autre exemple d'une séquence d'opération correspondant aux exemples de la figure 6, dans ce cas avec un ensemble de 7 registres de mémoire cache dans le matériel.

Avec référence à la figure 1, l'architecture principale d'un système de traitement est vu pour inclure un processeur central principal 11 et un moteur multiplicateur 13 partageant une mémoire à accès direct ou RAM 15 et une mémoire cache à registres de commande de multiplicateur 17. Le moteur multiplicateur 13 comprend un module de multiplication accumulation (MAC) 21, un séquenceur d'opérations 23 connecté pour envoyer des signaux de commande au MAC 21, des registres de commande 25 et des registres de données internes 27.

Des modules de gestion de mémoire/protection de mémoire (MMU/MPU) 14 et 19 assurent l'interface de la RAM 15 et de la mémoire cache 17 avec le processeur 11 et le moteur multiplicateur 13. Dans notre configuration préférée, il y a un MMU/MPU pour le processeur central 11 concernant à la fois les accès à la RAM et aux périphériques (afin de contrôler/limiter les accès à certaines zones/périphériques). Ici, le moteur multiplicateur 13 est considéré comme un périphérique. Comme le moteur multiplicateur a un accès direct à la RAM 15, il pourrait être un moyen pour l'utilisateur de surmonter les limitations d'accès spécifiées dans le MMU/MPU côté noyau 19. Par conséquent, nous proposons un

autre MMU/MPU 14 pour contrôler les accès à la mémoire par le moteur multiplicateur 13. Les deux modules MMU/MPU 14 et 19 doivent être configurés de manière cohérente, mais il n'y a aucun lien entre eux et leur fonctionnement est indépendant.

Le moteur multiplicateur 13 n'a généralement pas de ROM dédiée, mais il est configuré et paramétré par le processeur central 11. Les registres de commande 25 sont connectés à la mémoire cache des registres de commande 17 desquels ils reçoivent des instructions provenant du processeur central 11. Les registres de commande 25 transmettent les paramètres d'instruction et des informations d'état au séquenceur d'opérations 23, et communiquent aussi avec le module MAC 21, par exemple pour sélectionner le mode MAC (arithmétique standard ou arithmétique polynomiale) pour ces modules MAC qui peuvent être capables des deux types d'arithmétique, pour sélectionner une opération MAC à mot unique ou à mots multiples, et pour communiquer n'importe quelle valeur de report depuis une opération MAC en cours ou précédente.

Les registres d'adresses ou de données internes 27 sont reliées à la RAM partagée 15 pour recevoir et émettre des paramètres d'opérande d'une opération MAC. Le séquenceur d'opération 23 est de préférence préprogrammé par logiciel interne (firmware) selon l'invention décrite ici. Le séquenceur d'opérations 23 envoie des instructions et des adresses aux registres internes 27 et de là à la RAM partagée 15 pour diriger le chargement de segments d'opérandes d'une largeur d'un mot dans un ordre spécifié selon la présente invention. L'architecture est typiquement construite de sorte que, lorsque le moteur multiplicateur 13 fonctionne, il a un accès privilégié à

une partie spécifique de la RAM partagée 15. Cela permet au processeur central 11 d'accéder encore à d'autres parties de la RAM 15 pendant un calcul. En variante, l'accès à la RAM partagée 15 pourrait être entièrement dédié au moteur multiplicateur 13 pendant un calcul et accessible au processeur central 11 uniquement lorsque le multiplicateur 13 ne l'utilise pas.

Le module MAC 21 peut être basé sur une taille de mot de 32 bits. Dans ce cas, les longueurs d'opérande sont toujours des multiples de 4 octets et les opérandes sont alignés sur des limites de mot de 32 bits avec des zéros d'en-tête si nécessaire. Ce choix facilite les calculs d'adresses pour la RAM partagée 15 puisque le processeur 11 travaille typiquement sur des adresses en octets. Le module MAC 21 aurait une paire d'entrées d'opérandes X et Y (de largeur de mot) de 32 bits pour l'ensemble multiplicateur, une entrée d'accumulateur Z d'une largeur de deux mots, une sortie de résultats de multiplicateur d'une largeur de deux mots qui forme une seconde entrée d'accumulateur, et une sortie d'accumulateur R d'une largeur de deux mots.

Bien que le module MAC 21 ne fonctionne que sur des entrées X et Y d'une largeur d'un mot, le moteur multiplicateur 13 général, y compris le séquenceur d'opérations programmées 23, peut être considéré comme un multiplicateur de grands entiers (de plusieurs mots). Il prend en charge les opérations de multiplication efficaces telles que multiplication-accumulation de nombres à N mots, élévation au carré-accumulation de nombres de N mots (entrée multiplication Y = X), multiplication ou élévation au carré de nombres à N mots sans accumulation (entrée accumulateur Z = 0),

multiplication-accumulation d'un nombre de N mots par une constante A de 1 mot (et même d'1 octet).

Avec référence à la figure 2, l'interaction entre le moteur multiplicateur 13 avec ses divers registres et mémoires cache et avec la RAM partagée 15 est illustrée par un diagramme d'interface. La RAM 15 enregistre les paramètres X, Y, Z et R à des blocs d'adresse spécifiques qu'il faut pointer pour y accéder. Les mots des opérandes sont toujours enregistrés bit de moindre poids en premier dans la RAM, à commencer par une adresse de base. Pour demander un paramètre, ou un mot de segment d'opérande spécifique, le registre d'adresse correspondant X ADDR, Y ADDR, Z ADDR ou R ADDR (certains des registres internes 27 sur la figure 1) doit être chargé avec l'adresse associée. Les mots adressés sont ensuite chargés vers et depuis le registre de données correspondant RX, RY, RZ et RR (plus des registres internes 27 sur la figure 1 utilisés par le module MAC 21).

Les registres 25 incluent typiquement un ou plusieurs registres d'opérations pour spécifier une opération désirée à exécuter (multiplication, multiplication-accumulation, multiplication-soustraction, multiplication par une constante d'un seul mot, élévation au carré, etc.), un ou plusieurs registres spécifiant diverses options (mode naturel ou polynomial, opération entière ou tronquée, report ou entrée de report, etc.) ainsi qu'indications des longueurs d'opérande, et un ou plusieurs registres indiquant diverses conditions (occupé/inactif, un résultat de dépassement par valeurs supérieures/dépassement par valeurs inférieures/zéro, conditions d'erreur, etc.).

Avec référence à la figure 3, le module MAC 21 de la figure 1 est constitué d'un ensemble multiplicateur de nombres entiers 31 recevant des opérandes à mot unique ou plus généralement des segments de mot à mot unique d'opérandes plus grandes de plusieurs mots via des registres de données RX et RY chargés de la RAM partagée. L'ensemble multiplicateur 31 envoie un résultat à deux mots multipliant les mots d'entrée vers un accumulateur 33. L'accumulateur a une taille de trois mots (= 96 bits) plus un certain nombre de bits de report suffisant pour gérer une taille d'opérande maximum spécifiée. Par exemple, pour les opérandes de 512 mots, la colonne la plus longue a 512 lignes de produits intermédiaires à ajouter, nécessitant ainsi un espace de 9 bits pour la somme de report. Cela donne une largeur d'accumulateur totale de 105 bits pour cet exemple. L'accumulateur 33 reçoit aussi un paramètre d'entrée à deux mots provenant du registre de données RZ, et envoie un résultat à un registre de données de résultat RR. Un retour à deux mots peut être proposé depuis la sortie de l'accumulateur ou du registre de données RR à destination du registre d'opérandes d'accumulation pour permettre au segment d'opérande accumulé d'être mis à jour avant que le résultat final dans le registre RR ne soit écrit de nouveau dans la RAM. Une taille typique pour le module de multiplication accumulation (MAC) gère des mots de 32 bits, avec des entrées d'opérande de 32 bits provenant des registres RX et RY à destination de l'ensemble multiplicateur, et avec une sortie d'ensemble multiplicateur d'opérandes de 64 bits, et des entrées et sorties d'accumulateur vers et depuis des registres RZ et RR.

Soit A représentant un opérande d'un seul mot ou un segment d'opérande X de plusieurs mots qui est chargé dans l'ensemble multiplicateur 31 depuis le registre de données RX, et B représentant un segment d'un seul mot d'un opérande de plusieurs mots Y qui est chargé dans l'ensemble multiplicateur 31 depuis le registre de données RY, puis $A = \sum_i a_i 2^i$, $B = \sum_j b_j 2^j$, où a_i et b_j sont les bits individuels de l'opérande ou du segment, et où i et j sont compris entre 0 et 31.

Pour une multiplication sur le champ d'entier z_p,

$$AB = \sum_i \sum_j a_i b_j 2^{i+j}.$$

Pour une multiplication au-dessus du champ de Galois $GF(2^n)$, l'addition de deux bits est réduite modulo 2 de sorte que

$$AB = \sum_{k=0}^{2n-2} (2^k \cdot \sum_{i+j=k} (a_i \cdot b_j \bmod 2))$$

Il peut aussi y avoir un terme "report dans" W qui est ajouté. Le traitement du report dépend des options indiquées par les registres 25 mentionnés ci-dessus. Notez aussi que le terme "report dans" W ne doit pas nécessairement avoir un rapport direct au terme de report sortant provenant d'un calcul immédiatement antérieur. Finalement, l'opération du champ de Galois a une influence sur le traitement du report en ce que l'ajout du terme "report dans" W vers le bit de moindre poids est aussi conduit modulo 2.

Une multiplication grandeur nature d'opérandes de plusieurs mots X et Y implique une séquence des multiplications de mots uniques qui viennent d'être décrites. X et Y sont respectivement des nombres de N mots et de M mots. L'opération générale est

$R = [Z] \pm ((X \cdot Y) + W)$. Ceci peut être écrit sous forme de somme de produits comme :

$$R = [\Sigma_{k=0}^{N+M-1} Z_k b^k] \pm (\Sigma_{i=0}^{N-1} (\Sigma_{j=0}^{M-1} (X_i Y_j b^{i+j})) + W).$$

Cette formule est valide à la fois pour les opérations Z_p et $GF(2^n)$, et $b = 2^{32}$. L'accumulateur de reports de trois mots et plus peut calculer Acc := Acc $\pm(X_i \cdot Y_j)$ ou Acc := Acc $\pm(X_i \cdot Y_j \cdot 2^{32})$, si nécessaire. L'invention réside dans l'ordre particulier dans lequel les multiplications à un seul mot se déroulent.

Séquence d'opérations

En référence à la figure 4, le tableau montre une disposition d'opérations de multiplication mot par mot des premiers segments d'opérande B0-B7 de l'opérande à plusieurs mots Y. À titre d'illustration, les deux opérandes X et Y ont une largeur de B mots, mais ce ne doit pas nécessairement être le cas. Pour obtenir les mots de résultat R0 ... R15, il faudra éventuellement ajouter les divers produits partiels verticalement ensemble avec les mots d'accumulation correspondants C0 ... C15 de l'opérande Z. Dans l'ordre des accès à la mémoire, le chargement des segments d'opérandes, leur multiplication respective, et l'ajout aux segments d'accumulation, sont organisés en doubles colonnes de poids de résultat adjacents. Chaque groupe de deux colonnes est traité en partant soit du haut soit du bas et en progressant ligne par ligne en zigzag. Aussi, les groupes adjacents de paires de colonnes ne doivent pas nécessairement progresser dans la même direction. Ainsi, sur la figure 4, la séquence de multiplication pourrait aller ainsi : A1B0, A0B0, A0B1 ; A0B3, A0B2, A1B2, A1B1,

A3B1, A2B0, A3B0, A5B0, A4B0, A4B1, A3B1, A3B2, A2B2, A2B3, A1B3, A1B4, A0B4, A0B5 ; A0B7, A0B6, A1B6, A1B5, A2B5, A2B4, A3B4, A3B3, A4B3, A4B2, A5B2, A5B1, A6B1, A6B0, A7B0 ; A7B1, A7B2, A6B2, A6B3, A5B3, A5B4, A4B4, A4B5, A3B5, A3B6, A2B6, A2B7, A1B7 ; A3B7, A4B7, A4B6, A5B6, A5B5, A6B5, A6B4, A7B4, A7B3, A7B5, A7B6, A6B6, A6B7, A5B7 ; A7B7. Des points-virgules séparent les différents groupes de paires de colonnes dans la liste de séquence. Dans cet exemple particulier, des groupes successifs progressent dans des directions opposées (de haut en bas, puis de bas en haut, puis de haut en bas à nouveau, etc.) Notez que seul un opérande a besoin d'être lu avant que l'ensemble puisse exécuter la multiplication suivante, car un opérande est déjà en place depuis la multiplication suivante.

Si des groupes successifs de paires de colonnes progressent toujours dans la même direction, alors un certain nombre de registres de mémoire cache peut être fourni pour enregistrer certains segments d'opérande fréquemment utilisés nécessaires pour éviter deux lectures avant la première plusieurs multiplie au début de chaque groupe. Dans ce cas, cinq mémoires cache enregistreront A0, A1, B0, B1 et B2 pendant la moitié ascendante de la séquence de multiplication. La moitié droite de la figure 4, lorsque chaque groupe successif deviendra plus long.

Les figures 5a, 5b et 5c illustrent l'intérêt d'avoir de telles mémoires cache, en montrant une séquence de multiplication pour ce mode de réalisation de l'invention. La première colonne représente les accès de lecture de la RAM, la deuxième colonne représente des opérations de l'ensemble multiplicateur, et les cinq

colonnes les plus à droite montrent le contenu des cinq mémoires cache pendant chaque cycle de multiplication. Pour une multiplication de 8 mots par 8 mots avec 5 mémoires cache, il y a 64 cycles de multiplication, 7 cycles de chargement de mémoire cache uniquement, 5 cycles de lecture d'autres mémoires, plus 3 cycles à la fin pour terminer pour un total de seulement 79 cycles. L'avantage par rapport aux séquences précédentes est encore plus évident quand on multiplie des nombres entiers plus grands courants dans les applications cryptographiques (par exemple, 1024 mots par 1024 bits).

La taille en mots de chaque opérande peut être paire ou impaire, et ne doit pas nécessairement être la même pour les deux opérandes. Le cas présenté est souvent qualifié d'opération de "multiplication rectangulaire". Un exemple d'opération de multiplication et d'accumulation rectangulaire où un des opérandes X a un nombre impair de mots (5) et l'autre opérande de multiplication Y a un nombre pair de mots (14), et où l'opérande d'accumulation Z et le résultat R ont un nombre impair de mots (19) est présenté dans les figures 6, 7a, 7b et 7c à titre d'illustration. Dans cet exemple, il y a aussi 7 registres de mémoire cache internes pour enregistrer les segments de paramètre fréquemment utilisés. Seule une partie de la séquence totale est montrée, pour les groupes R0-R1, R2-R3, R4-R5, R6-R7, ... (R9 à R11 omis), R12-R13, R14-R15, et R16-R17. Dans les groupes, le modèle de multiplication encore une fois est arrangé de sorte que au plus un accès de lecture de la RAM et un accès d'écriture dans la RAM sont requis. La première colonne indique l'adresse RAM spécifique qui est appliquée et accédée dans des cycles successifs. Les

écritures dans la RAM sont représentées avec des bordures en gras. La deuxième et la troisième colonnes montrent les opérations exécutées par l'ensemble multiplicateur et l'accumulateur, respectivement. Les sept colonnes les plus à droite montrent le contenu des registres de mémoire cache internes utilisés par l'ensemble multiplicateur et l'accumulateur.

Cet exemple montre également que la séquence de multiplication ne doit pas nécessairement descendre strictement en zigzag de haut en bas ou remonter en zigzag de bas en haut sur tout un groupe de paires de colonnes. Au contraire, les débuts et/ou les fins de telles séquences en zigzag peuvent être déplacés et même évoluer dans une direction opposée par rapport à la première partie d'une séquence de groupe, parce que les paramètres nécessaires sont disponibles dans les mémoires cache. Ainsi, dans le premier groupe R0-R1, l'ordre peut être Y0X0, Y0X1, Y1X0, au lieu de Y0X0, Y1X0, Y0X1 pour une séquence strictement descendante, ou au lieu de Y0X1, Y0X0, Y1X0 pour une séquence strictement ascendante. Dans les deuxième, troisième et quatrième groupes, la séquence commence au milieu à Y1X1, Y3X1 et Y5X1 respectivement, descend en zigzag jusqu'en bas, puis remonte directement jusqu'à Y2X1, Y4X1 et Y6X1 respectivement et se termine en zigzag ascendant jusqu'en haut. Cela est possible parce que le segment d'opérande X1 est enregistré dans la mémoire cache interne, de sorte qu'une seule autre lecture suffit malgré le saut de Y0X3 à Y2X1, de Y0X4 à Y4X1, et de Y2X4 à Y6X1 respectivement au milieu de la séquence. Cela est aussi valable pour l'un quelconque des groupes omis dans les figures 7a-7c et pour le groupe R12-R13. Dans le groupe R14-R15, la séquence commence en

haut et descend strictement en zigzag jusqu'à Y12X3, puis saute à Y10X4 et se termine en zigzag ascendant avec Y11X4 et Y11X3. Ce saut et changement de direction dans la séquence est permis parce que le paramètre X4 est déjà disponible dans un registre de la mémoire cache. Le groupe de multiplication final R16-R17 est représenté dans un modèle en zigzag strictement descendant, mais peut se dérouler dans n'importe quel ordre parce que les segments d'opérande X3 et X4 sont disponibles dans des registres de la mémoire cache.

Les opérations d'élévation au carré et les opérations d'élévations au carré-accumulation se déroulent de la même manière que l'un quelconque des exemples précédents, sauf que les opérandes X et Y sont identiques. Cependant, comme les segments spécifiques de X et Y ne sont généralement pas les mêmes dans un cycle de multiplication particulier quelconque, les opérations d'élévation au carré peuvent être traitées tout comme n'importe quelle autre opération de multiplication dans laquelle les opérandes X et Y diffèrent.

REVENDICATIONS

1. Procédé de fonctionnement d'un circuit de multiplication pour calculer le produit de deux opérandes (X et Y) dont au moins un d'eaux est sensiblement plus large que le circuit de multiplication, le circuit de multiplication ayant une paire d'entrées d'opérandes d'une largeur de deux mots et une sortie résultat d'une largeur de deux mots, où un mot est un nombre spécifié de bits, chacun des opérandes composé d'une pluralité de segments d'opérande contigus d'une largeur de mot ordonnée (x_k et y_m), caractérisés par des poids spécifiques (k et m), le circuit de multiplication ayant accès à une mémoire, le procédé comprenant les étapes consistant à :

charger des segments d'opérandes d'une largeur d'un mot des deux opérandes dans un ordre spécifié à partir de la mémoire dans le circuit de multiplication, le circuit de multiplication comprenant au moins deux registres (RX et RY) ayant accès à ladite mémoire pour contenir temporairement les segments chargés ;

multiplier les segments chargés pour obtenir des produits intermédiaires d'une largeur de deux mots, les produits intermédiaires ayant un poids égal à la somme des poids des segments chargés ;

ajouter des produits intermédiaires du même poids dans un accumulateur (33), l'accumulateur ayant une taille de trois mots plus un nombre de bits de report, suffisante pour gérer une taille d'opérande maximale spécifiée, l'accumulateur connecté à un registre d'entrée de deux mots (RZ) pour contenir temporairement tout produit d'un poids spécifique ajouté précédemment et un registre de

sortie à deux mots (RR) pour contenir les résultats d'une étape d'addition, lesdits registres (RZ et RR) ayant accès à ladite mémoire ; et

enregistrer les résultats accumulés depuis ledit registre de sortie (RR) de nouveau dans ladite mémoire au moins après accumulation de tous les produits intermédiaires du poids spécifié ;

dans lequel l'ordre spécifié pour les segments d'opérande de chargement dans lesdits registres est une séquence définie par les poids des produits intermédiaires résultants, dans lequel l'étape de multiplication est faite par groupes successifs de deux poids de produits adjacents, avec la séquence à l'intérieur d'un groupe sélectionnée de sorte que, à l'exception d'une première opération de multiplication dans un groupe donné, au plus un des segments d'opérande doive être lu de la mémoire et placé dans un registre (RX ou RY), l'autre segment d'opérande étant déjà enregistré dans l'autre registre (RY ou RX) à partir de l'opération de multiplication immédiatement précédente.

2. Procédé de la revendication 1, dans lequel le registre d'entrée de l'accumulateur (RZ) est chargé avec un segment d'un mot accumulé (Z), où le segment est du même poids que les produits intermédiaires à former dans ladite étape de multiplication, moyennant quoi le procédé de fonctionnement exécute une opération de multiplication-accumulation sur une paire d'opérandes de multiplication et un opérande d'accumulation.

3. Procédé de la revendication 1, dans lequel la séquence se déroule à l'intérieur d'un groupe dans un modèle en

zigzag de poids de segments d'opérande augmentant constamment.

4. Procédé de la revendication 1, dans lequel la séquence se déroule à l'intérieur d'un groupe dans un modèle en zigzag de poids de segments d'opérande diminuant constamment.

5. Procédé de la revendication 1, dans lequel le circuit de multiplication contient en outre un ensemble de registre de mémoire cache interne, lesdits registres de mémoire cache (17) étant chargés depuis la mémoire avec des segments d'opérande qui sont fréquemment utilisés dans une opération de multiplication.

6. Procédé de la revendication 1, dans lequel la séquence se déroule à l'intérieur d'un groupe commençant par une opération de multiplication avec au moins un segment d'opérande enregistré dans un registre de mémoire cache (17) dans une première direction en zigzag de poids de segments d'opérande augmentant ou diminuant constamment, puis passe directement à une opération de multiplication du groupe pas encore exécutée qui a aussi au moins un segment d'opérande enregistré dans un registre de mémoire cache et continue dans une deuxième direction en zigzag de poids de segments d'opérande diminuant ou augmentant constamment jusqu'à ce que toutes les opérations de multiplication pour ce groupe soient terminées.

7. Procédé de la revendication 1, dans lequel l'ordre spécifié est préprogrammé dans le logiciel interne dans un séquenceur d'opérations du circuit de multiplication,

avec des longueurs de mot d'opérande incluses en tant que paramètre d'entrée pour une commande de multiplication.

8. Circuit de multiplication, comprenant :

un module de multiplication-accumulation (MAC) (21) comprenant un ensemble multiplicateur qui introduit des segments d'opérandes récepteur à mot unique à multiplier pour former un produit intermédiaire de deux mots, et comprenant aussi un circuit accumulateur avec une première entrée de deux mots pour recevoir des produits intermédiaires provenant du multiplicateur, une deuxième entrée de deux mots pour recevoir une valeur accumulée, et une sortie de trois mots plus un certain nombre de bits de report suffisant pour gérer une taille d'opérande maximum spécifiée pour proposer une somme de valeurs d'entrée provenant des entrées de deux mots, la sortie de l'accumulateur retournant aussi deux mots à la deuxième entrée de deux mots ;

un ensemble de registres d'adresses internes pour adresser une mémoire à accès direct ;

un ensemble de registres de données internes (RX, RY, RZ, RR) connectés pour recevoir et transmettre des segments desdites opérandes depuis et à destination de ladite mémoire et aussi connectés audit ensemble multiplicateur et audit accumulateur de manière à fournir des segments d'opérande aux entrées de ceux-ci et à en recevoir des résultats, chacun desdits segments d'opérande ayant un poids spécifié ; et

un séquenceur d'opérations pour contrôler l'accès à la mémoire par lesdits adresse interne et registres de données et contrôler la séquence d'opérations de

multiplication et d'accumulation par lesdits ensemble multiplicateur et accumulateur respectifs,

dans lequel ladite séquence est définie par les poids des produits résultants égaux à la somme des poids des segments d'opérande en cours de multiplication, dans lequel les opérations de multiplication sont faites dans des groupes successifs de deux poids de produits adjacents, avec la séquence dans un groupe sélectionné de sorte que, excepté une première opération de multiplication dans un groupe donné, au maximum un des segments d'opérande doive être lu de la mémoire dans un registre (RX ou RY), l'autre segment d'opérande étant déjà enregistré dans l'autre registre (RY ou RX) depuis l'opération de multiplication immédiatement précédente.

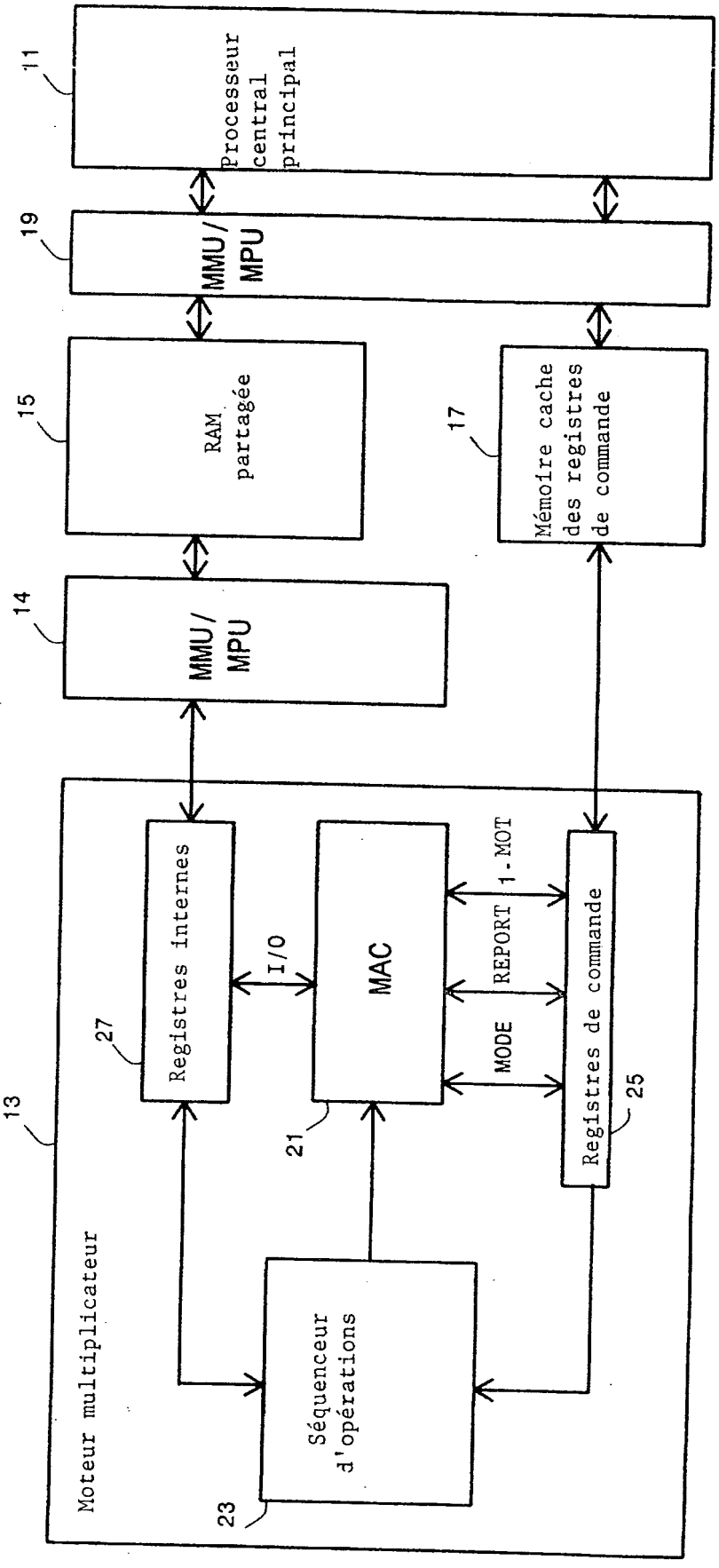
11
Processeur central principal
19
MMU/ MPU
15
RAM partagée
17
Mémoire cache des registres de commande
14
MMU/ MPU
13
Moteur multiplicateur
27
Registres internes
I/O
21
MAC
1-MOT
REPORT
MODE
25
Registres de commande
23
Séquenceur d'opérations

Fig. 1

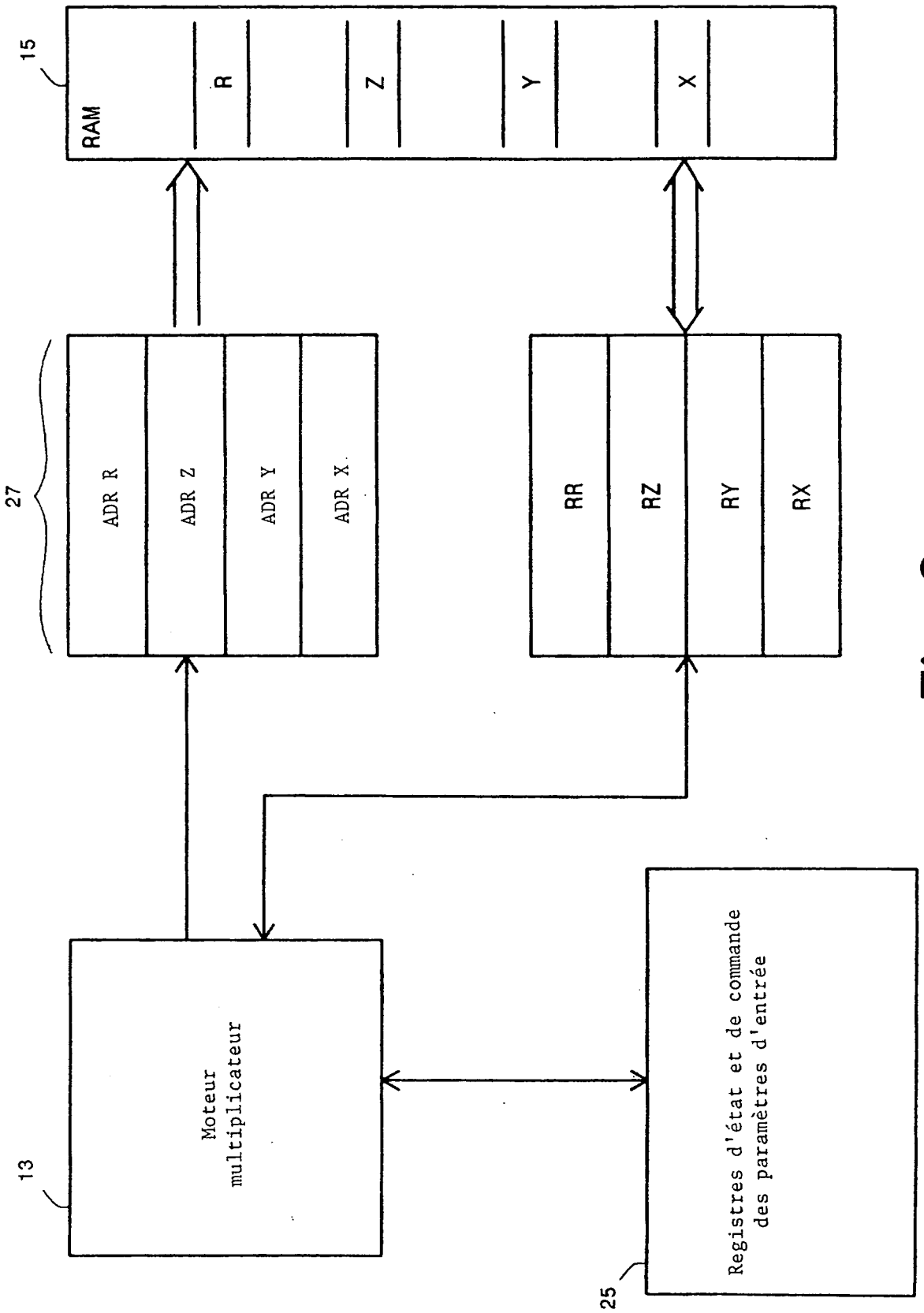
15
RAM
R
Z
Y
X
27
ADR R
ADR Z
ADR Y
ADR X
RR
RZ
RY
RX
13
Moteur multiplicateur
25
Registres d'état et de commande des paramètres d'entrée

Fig 2

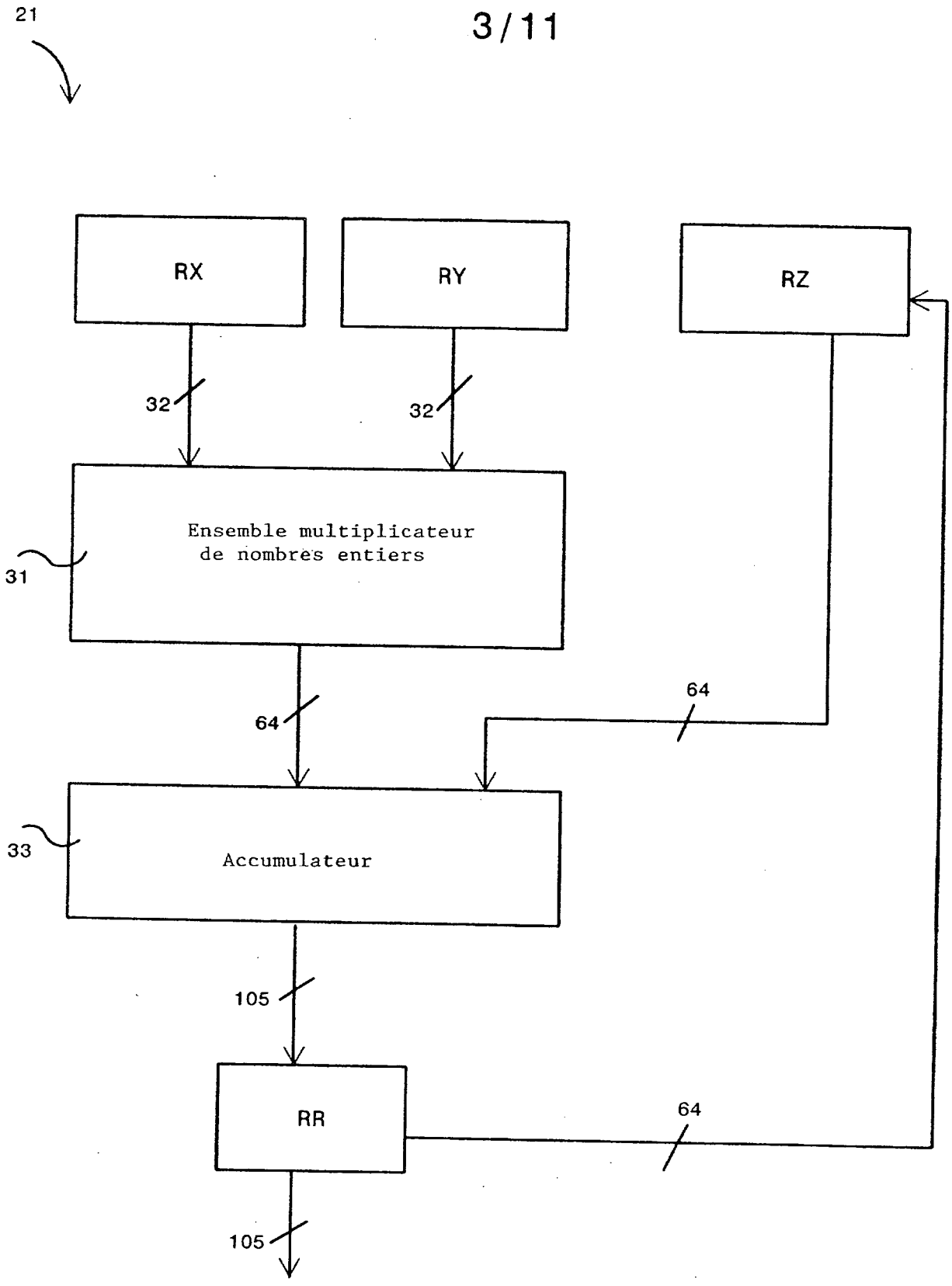
21
RX
RY
RZ
32
32
Ensemble multiplicateur
de nombres entiers
31
64
64
33
Accumulateur
105
RR
64
105

Fig. 3

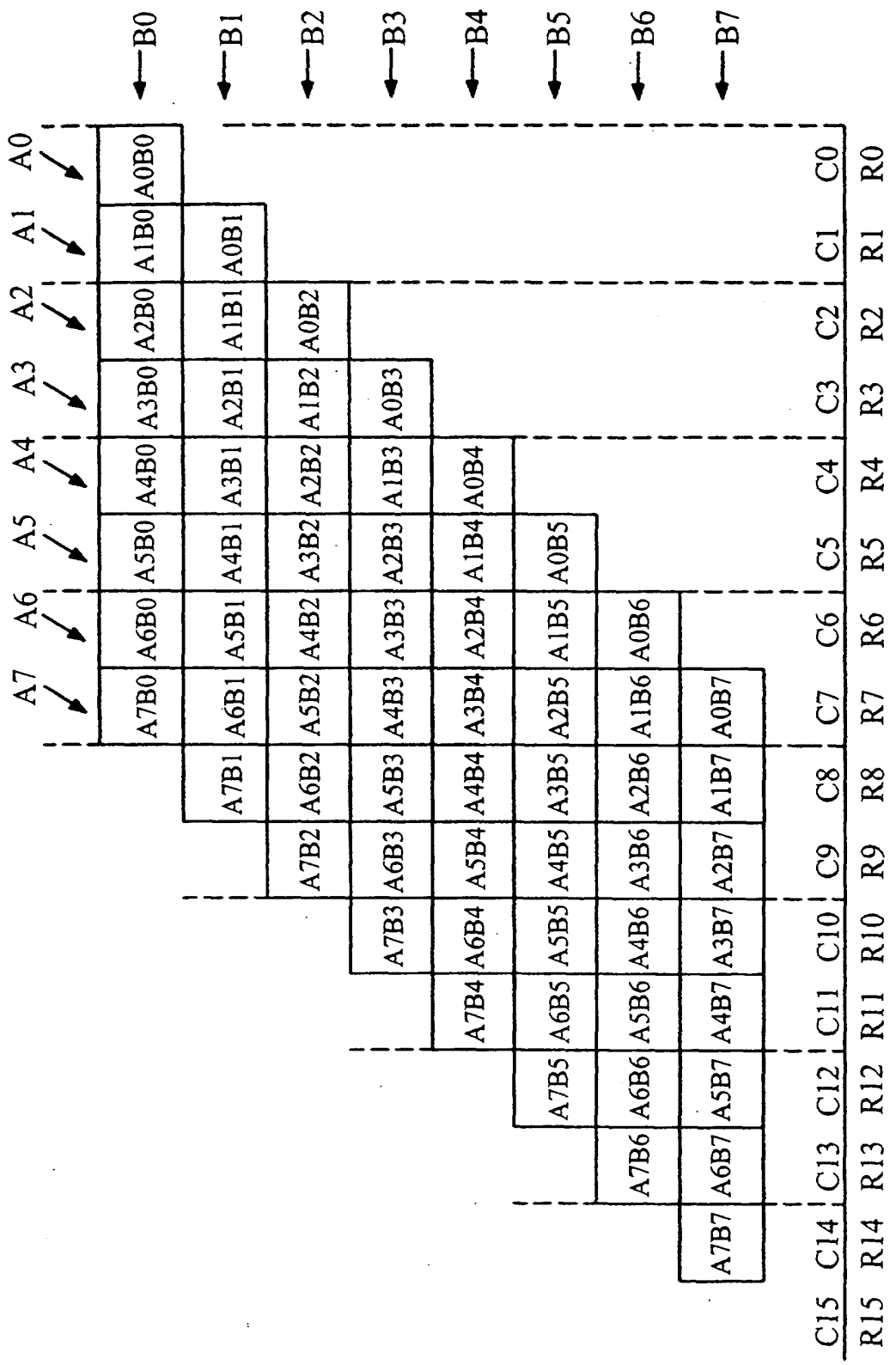
B0
B1
B2
B3
B4
B5
B6
B7
A0
A1
A2
A3
A4
A5
A6
A7
A0B0
A1B0 A0B1
A2B0 A1B1 A0B2
A3B0 A2B1 A1B2 A0B3
A4B0 A3B1 A2B2 A1B3 A0B4
A5B0 A4B1 A3B2 A2B3 A1B4 A0B5
A6B0 A5B1 A4B2 A3B3 A2B4 A1B5 A0B6
A7B0 A6B1 A5B2 A4B3 A3B4 A2B5 A1B6 A0B7
A7B1 A6B2 A5B3 A4B4 A3B5 A2B6 A1B7
A7B2 A6B3 A5B4 A4B5 A3B6 A2B7
A7B3 A6B4 A5B5 A4B6 A3B7
A7B4 A6B5 A5B6 A4B7
A7B5 A6B6 A5B7
A7B6 A6B7
A7B7
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14 C15
R0 R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13 R14 R15

Fig 4

RAM	MUL	C_A0	C_A1	C_B0	C_B1	C_B2
A0						
A1		A0				
B0		A0	A1			
B1		A0	A1	B0		
B2		A0	A1	B0	B1	
	A1*B0	A0	A1	B0	B1	B2
	A0*B0	A0	A1	B0	B1	B2
	A0*B1	A0	A1	B0	B1	B2
A3		A0	A1	B0	B1	B2
A2	A3*B0	A0	A1	B0	B1	B2
	A2*B0	A0	A1	B0	B1	B2
	A2*B1	A0	A1	B0	B1	B2
	A1*B1	A0	A1	B0	B1	B2
	A1*B2	A0	A1	B0	B1	B2
B3	A0*B2	A0	A1	B0	B1	B2
	A0*B3	A0	A1	B0	B1	B2
A5		A0	A1	B0	B1	B2
A4	A5*B0	A0	A1	B0	B1	B2
	A4*B0	A0	A1	B0	B1	B2
A3	A4*B1	A0	A1	B0	B1	B2
	A3*B1	A0	A1	B0	B1	B2
A2	A3*B2	A0	A1	B0	B1	B2
B3	A2*B2	A0	A1	B0	B1	B2
	A2*B3	A0	A1	B0	B1	B2
B4	A1*B3	A0	A1	B0	B1	B2
	A1*B4	A0	A1	B0	B1	B2
B5	A0*B4	A0	A1	B0	B1	B2
	A0*B5	A0	A1	B0	B1	B2

Fig 5A

RAM	MUL	C_A0	C_A1	C_B0	C_B1	C_B2
A7		A0	A1	B0	B1	B2
A6	A7*B0	A0	A1	B0	B1	B2
	A6*B0	A0	A1	B0	B1	B2
A5	A6*B1	A0	A1	B0	B1	B2
	A5*B1	A0	A1	B0	B1	B2
A4	A5*B2	A0	A1	B0	B1	B2
B3	A4*B2	A0	A1	B0	B1	B2
A3	A4*B3	A0	A1			
B4	A3*B3	A0	A1			
A2	A3*B4	A0	A1			
B5	A2*B4	A0	A1			
	A2*B5	A0	A1	B5		
B6	A1*B5	A0	A1	B5		
	A1*B6	A0	A1	B5	B6	
B7	A0*B6	A0	A1	B5	B6	
	A0*B7	A0	A1	B5	B6	B7
A7				B5	B6	B7
A6			A7	B5	B6	B7
B1		A6	A7	B5	B6	B7
B2	A7*B1	A6	A7	B5	B6	B7
	A7*B2	A6	A7	B5	B6	B7
B3	A6*B2	A6	A7	B5	B6	B7
A5	A6*B3	A6	A7	B5	B6	B7
B4	A5*B3	A6	A7	B5	B6	B7
A4	A5*B4	A6	A7	B5	B6	B7
	A4*B4	A6	A7	B5	B6	B7
A3	A4*B5	A6	A7	B5	B6	B7
	A3*B5	A6	A7	B5	B6	B7
A2	A3*B6	A6	A7	B5	B6	B7
	A2*B6	A6	A7	B5	B6	B7
A1	A2*B7	A6	A7	B5	B6	B7
	A1*B7	A6	A7	B5	B6	B7

Fig 5B

RAM	MUL	C_A0	C_A1	C_B0	C_B1	C_B2
B3		A6	A7	B5	B6	B7
B4	A7*B3	A6	A7	B5	B6	B7
	A7*B4	A6	A7	B5	B6	B7
	A6*B4	A6	A7	B5	B6	B7
A5	A6*B5	A6	A7	B5	B6	B7
	A5*B5	A6	A7	B5	B6	B7
A4	A5*B6	A6	A7	B5	B6	B7
	A4*B6	A6	A7	B5	B6	B7
A3	A4*B7	A6	A7	B5	B6	B7
	A3*B7	A6	A7	B5	B6	B7
	A7*B5	A6	A7	B5	B6	B7
	A7*B6	A6	A7	B5	B6	B7
	A6*B6	A6	A7	B5	B6	B7
A5	A6*B7	A6	A7	B5	B6	B7
	A5*B7	A6	A7	B5	B6	B7
	A7*B7	A6	A7	B5	B6	B7

Fig. 5C

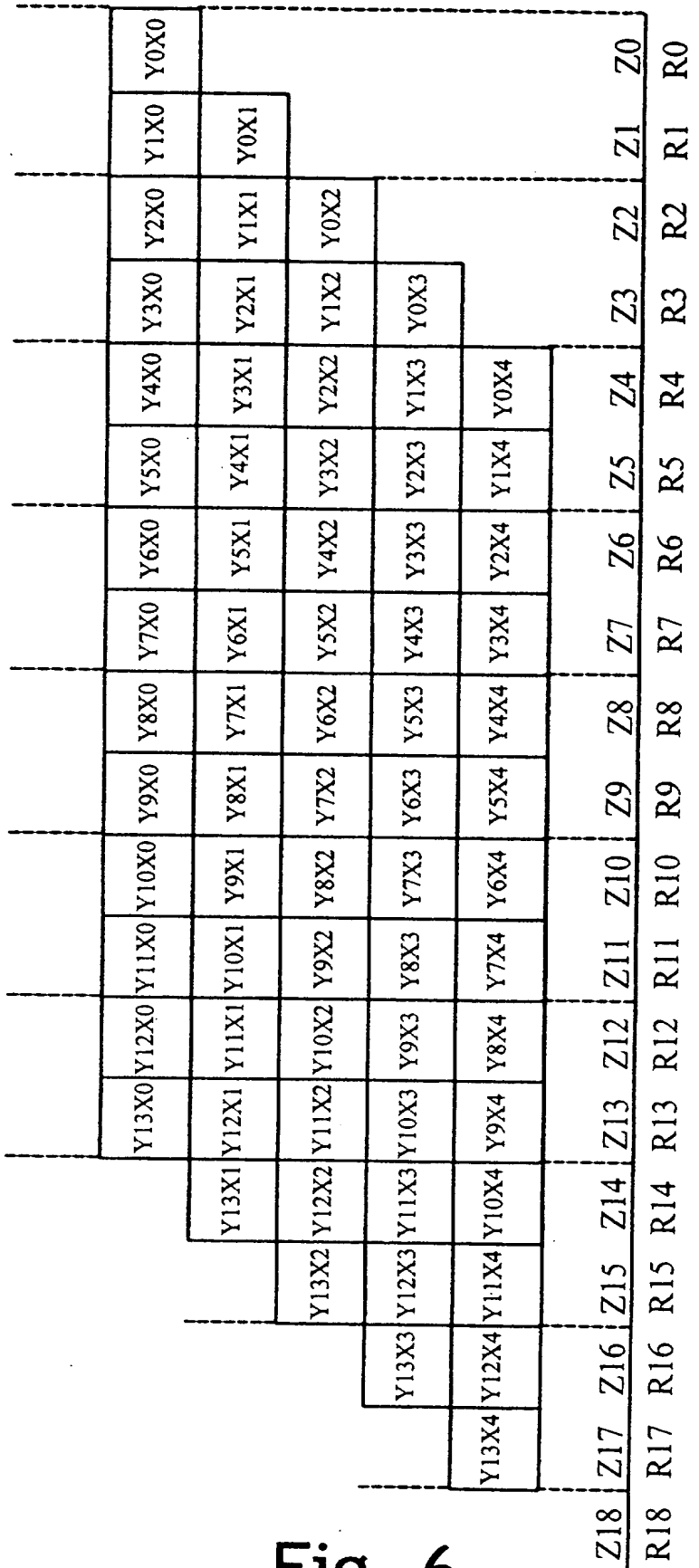
Y0X0
Y1X0 Y0X1
Y2X0 Y1X1 Y0X2
Y3X0 Y2X1 Y1X2 Y0X3
Y4X0 Y3X1 Y2X2 Y1X3 Y0X4
Y5X0 Y4X1 Y3X2 Y2X3 Y1X4
Y6X0 Y5X1 Y4X2 Y3X3 Y2X4
Y7X0 Y6X1 Y5X2 Y4X3 Y3X4
Y8X0 Y7X1 Y6X2 Y5X3 Y4X4
Y9X0 Y8X1 Y7X2 Y6X3 Y5X4
Y10X0 Y9X1 Y8X2 Y7X3 Y6X4
Y11X0 Y10X1 Y9X2 Y8X3 Y7X4
Y12X0 Y11X1 Y10X2 Y9X3 Y8X4
Y13X0 Y12X1 Y11X2 Y10X3 Y9X4
Y13X1 Y12X2 Y11X3 Y10X4
Y13X2 Y12X3 Y11X4
Y13X3 Y12X4
Y13X4
Z0 Z1 Z2 Z3 Z4 Z5 Z6 Z7 Z8 Z9 Z10 Z11 Z12 Z13 Z14 Z15 Z16 Z17 Z18
R0 R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13 R14 R15 R16 R17 R18

Fig 6

RW1	MUL1	ADD	y0	y1	y2	x0	x1	x2	x3
Y0									
Z0			Y0						
X0			Y0		Z0				
Z1	Y0X0	Z0	Y0		Z0	X0			
X1			Y0		Z1	X0			
Y1	Y0X1	Z1	Y0		Z1	X0	X1		
Z2	Y1X0		Y0	Y1	Z1	X0	X1		
X2			Y0	Y1	Z2	X0	X1		
R0	Y1X1	Z2	Y0	Y1	Z2	X0	X1	X2	
R1			Y0	Y1	Z2	X0	X1	X2	
Z3	Y1X2		Y0	Y1	Z2	X0	X1	X2	
X3	Y0X2		Y0	Y1	Z3	X0	X1	X2	
Y2	Y0X3	Z3	Y0	Y1	Z3	X0	X1	X2	X3
X0	Y2X1		Y2	Y1	Z3	X0	X1	X2	X3
Y3	Y2X0		Y2	Y1	Z3	X0	X1	X2	X3
Z4	Y3X0		Y2	Y3	Z3	X0	X1	X2	X3
Z5	Y3X1	Z4	Y2	Y3	Z4	X0	X1	X2	X3
R2	Y3X2	Z5	Y2	Y3	Z5	X0	X1	X2	X3
R3	Y2X2		Y2	Y3	Z5	X0	X1	X2	X3
Y1	Y2X3		Y2	Y3	Z5	X0	X1	X2	X3
X4	Y1X3		Y1	Y3	Z5	X0	X1	X2	X3
Y0	Y1X4		Y1	Y3	Z5	X4	X1	X2	X3
Y4	Y0X4		Y0	Y3	Z5	X4	X1	X2	X3

Fig. 7A

RW1	MUL1	ADD	y0	y1	y2	x0	x1	x2	x3
X0	Y4X1		Y4	Y3	Z5	X4	X1	X2	X3
Y5	Y4X0		Y4	Y3	Z5	X0	X1	X2	X3
Z6	Y5X0		Y4	Y5	Z5	X0	X1	X2	X3
Z7	Y5X1	Z6	Y4	Y5	Z6	X0	X1	X2	X3
R4	Y5X2	Z7	Y4	Y5	Z7	X0	X1	X2	X3
R5	Y4X2		Y4	Y5	Z7	X0	X1	X2	X3
Y3	Y4X3		Y4	Y5	Z7	X0	X1	X2	X3
X4	Y3X3		Y3	Y5	Z7	X0	X1	X2	X3
Y2	Y3X4		Y3	Y5	Z7	X4	X1	X2	X3
Y6	Y2X4		Y2	Y5	Z7	X4	X1	X2	X3
X0	Y6X1		Y6	Y5	Z7	X4	X1	X2	X3
Y7	Y6X0		Y6	Y5	Z7	X0	X1	X2	X3
Z8	Y7X0		Y6	Y7	Z7	X0	X1	X2	X3
	Y11X1	Z12	Y10	Y11	Z12	X0	X1	X2	X3
	Y11X2	Z13	Y10	Y11	Z13	X0	X1	X2	X3
	Y10X2		Y10	Y11	Z13	X0	X1	X2	X3
Y4	Y10X3		Y10	Y11	Z13	X0	X1	X2	X3
X11	Y9X3		Y10	Y9	Z13	X0	X1	X2	X3
Y3	Y9X4		Y10	Y9	Z13	X4	X1	X2	X3
Y14	Y8X4		Y10	Y8	Z13	X4	X1	X2	X3
X0	Y12X1		Y10	Y12	Z13	X4	X1	X2	X3
Y15	Y12X0		Y10	Y12	Z13	X0	X1	X2	X3
Z14	Y13X0		Y10	Y12	Z13	X0	X1	X2	X3

Fig 7B

RW1	MUL1	ADD	y0	y1	y2	x0	x1	x2	x3
Z15	Y13X1	Z14	Y10	Y12	Y13	Z14	X1	X2	X3
R12	Y13X2	Z15	Y10	Y12	Y13	Z15	X1	X2	X3
R13	Y12X2		Y10	Y12	Y13	Z15	X1	X2	X3
X4	Y12X3		Y10	Y12	Y13	Z15	X1	X2	X3
Y11	Y10X4		Y10	Y12	Y13	Z15	X4	X2	X3
Z16	Y11X4		Y11	Y12	Y13	Z15	X4	X2	X3
	Y11X3		Y11	Y12	Y13	Z16	X4	X2	X3
Z17	Y13X3	Z16	Y11	Y12	Y13	Z16	X4	X2	X3
Z18	Y13X4	Z17	Y11	Y12	Y13	Z17	X4	X2	X3
R14	Y12X4		Y11	Y12	Y13	Z18	X4	X2	R15
R15			Y11	Y12	Y13	Z18	X4	X2	R15
		Z18	Y11	Y12	Y13	Z18	X4	X2	R15
R16			Y11	Y12	Y13	Z18	X4	X2	R15
R17			Y11	Y12	Y13	Z18	X4	X2	R15
R18			Y11	Y12	Y13	Z18	X4	X2	R15

Fig.7C

reçue le 13/05/03

INPI
INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

BREVET D'INVENTION
CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle - Livre VI

cerfa N° 11235*03

DÉPARTEMENT DES BREVETS
26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 33 (1) 53 04 53 04 Télécopie : 33 (1) 42 94 86 54

DÉSIGNATION D'INVENTEUR(S) Page N° 1../1..

INV

(À fournir dans le cas où les demandeurs et les inventeurs ne sont pas les mêmes personnes)

Cet imprimé est à remplir lisiblement à l'encre noire

DB 113 @ W / 270601

Vos références pour ce dossier *(facultatif)*	33312/FR
N° D'ENREGISTREMENT NATIONAL	0304299

TITRE DE L'INVENTION (200 caractères ou espaces maximum)

SÉQUENCE DE MULTIPLICATION EFFICACE POUR OPÉRANDES À GRANDS NOMBRES ENTIERS PLUS LARGES QUE LE MATÉRIEL MULTIPLICATEUR

LE(S) DEMANDEUR(S) :

Atmel Corporation
2325 Orchard Parkway
SAN JOSE California 95131
U.S.A.

DESIGNE(NT) EN TANT QU'INVENTEUR(S) :

1 Nom		DUPAQUIS
Prénoms		Vincent
Adresse	Rue	4 rue du Collet
	Code postal et ville	13124 PEYPIN
Société d'appartenance *(facultatif)*		
2 Nom		PARIS
Prénoms		Laurent
Adresse	Rue	28 Le Ribas
	Code postal et ville	13790 ROUSSET
Société d'appartenance *(facultatif)*		
3 Nom		
Prénoms		
Adresse	Rue	
	Code postal et ville	
Société d'appartenance *(facultatif)*		

S'il y a plus de trois inventeurs, utilisez plusieurs formulaires. Indiquez en haut à droite le N° de la page suivi du nombre de pages.

DATE ET SIGNATURE(S)
DU (DES) DEMANDEUR(S)
OU DU MANDATAIRE
(Nom et qualité du signataire)

Le 07/04/2003

BRESSE Pierre 92-1038

La loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés s'applique aux réponses faites à ce formulaire. Elle garantit un droit d'accès et de rectification pour les données vous concernant auprès de l'INPI.